241-TP-002-001

Plan for Achieving Required ECS Throughput Performance

Technical Paper

August 1997

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

Nicholas C. Singer /s/ 8/25/97

Nicholas C. Singer, Principal Scientist/Engineer Date
EOSDIS Core System Project

SUBMITTED BY

Paul Fingerman /s/ 8/26/97

Paul Fingerman, SMO Manager Date
EOSDIS Core System Project

Technical Paper--Not intended for formal review or government approval.

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

This Technical Paper defines the plan for assuring that the EOSDIS Core System (ECS) will meet system throughput and response time requirements. It provides an overview of ECS and its performance requirements, and discusses current and future performance measurement and enhancement activities necessary to ensure that required ECS throughput performance is achieved. It also documents past ECS performance engineering activities.

Keywords: Performance, throughput, response time, system, ingest, benchmarking, tuning, modeling.

This page intentionally left blank.

Contents

Abstract

1. Introduction

2. Plan for Performance Measurement and Enhancement Activities

3. Schedule

Figures

Tables

Appendix A. ECS Performance Engineering Background

Appendix B. Performance Scenario Plan Template

Appendix C. Ingest-to-Archive Performance Scenario Plan

Abbreviations and Acronyms

1. Introduction

1.1 Purpose

This Technical Paper defines the plan for assuring that the EOSDIS Core System (ECS) will meet system throughput and response time requirements.

1.2 Scope

The scope is restricted to the Science and Data Processing Segment (SDPS) and the Communications and System Management Segment (CSMS). Performance of the Flight Operations Segment (FOS) is not considered here.

1.3 Organization

This paper is organized as follows:

Section 1 – Describes the purpose, organization, review status, and points of contact.

Section 2 – Provides an overview of ECS and discusses current and future performance measurement and enhancement activities necessary to ensure required ECS throughput performance.

Section 3 – Lays out the schedule for performance engineering activities between now and the Landsat-7 and AM-1 launches.

Appendix A – Documents ECS performance engineering activities through July 31, 1997.

Appendix B – Contains a template for Performance Scenario Plans.

Appendix C – Contains a filled-in Performance Scenario Plan for Ingest-to-Archive.

1.4 Review and Approval

This Technical Paper is an informal document approved at the Office Manager level. It does not require formal Government review or approval; however, it is submitted with the intent that review and comments will be forthcoming.

Questions regarding technical information contained within this Paper should be addressed to the following ECS and/or Goddard Space Flight Center (GSFC) contacts:

- ECS Contact
  - Nicholas C. Singer; Principal Scientist/Engineer and Manager, ECS Modeling and Performance Engineering; (301) 925-0520; nsinger@eos.hitc.com

- GSFC Contact
  - Chris Wilkinson; Earth Science Data and Information System (ESDIS) Systems Engineer; chris.wilkinson.1@gsfc.nasa.gov; (301) 614-5365.

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Systems
1616 McCormick Drive
Upper Marlboro, MD 20774

2. Plan for Performance Measurement and Enhancement Activities

2.1 Overview

2.1.1 ECS Basics

ECS is a very large, complex system, with potentially hundreds or even thousands of concurrent users. ECS will produce 260 standard data products on an ongoing basis. On a daily basis, ECS must process over 480 gigabytes of raw data and produce 18,000 product instances with an aggregate size of 1,600 gigabytes. Thus, ECS data archives will grow at the rate of two terabytes of data per day and are expected to reach a petabyte in size by 1999. In addition to standard production, ECS supports on-demand production of certain products in response to user requests. The planning and scheduling functions must be able to interleave on-demand production requests with standard production in a manner that maximizes computing resources and maintains the overall production schedule.

Figure 2.1-1 shows the major subsystems of an ECS distributed active archive center (DAAC) and the major data flows between subsystems. The subsystems are:

CLIENT—Provides the “client” part of the “Client / Server” access paradigm through graphical user interface and data/service access tools, as well as application program interface (API) libraries to ECS services.

INTEROPERABILITY—Provides application-level routing which facilitates dynamic client access to services and providers holding data collections and services.

DATA MANAGEMENT—Provides system-wide distributed search and access services with multiple science discipline views of data collections and “one stop shopping” with location transparent access to those services and data.

DATA SERVER—Provides search, access, archive repository, and distribution services with a science discipline view of data collections and an extensible Earth Science Data Type (ESDT) and Computer Science Data Type view of the archive holdings.

INGEST—Provides for the importation of data (raw data, science products, ancillary, correlative, documents, etc.) into ECS data repositories (Data Servers) on an ad hoc or scheduled basis.

PLANNING—Provides for pre-planning of routine/ad hoc/on-demand science data processing as well as management functions for handling deviations from the operations plans for individual DAAC sites.

DATA PROCESSING—Provides the functions to host science algorithm software, perform science software integration and test, production data processing, process resource management, and includes facilities and toolkits which offer true software portability across advanced computing platforms.

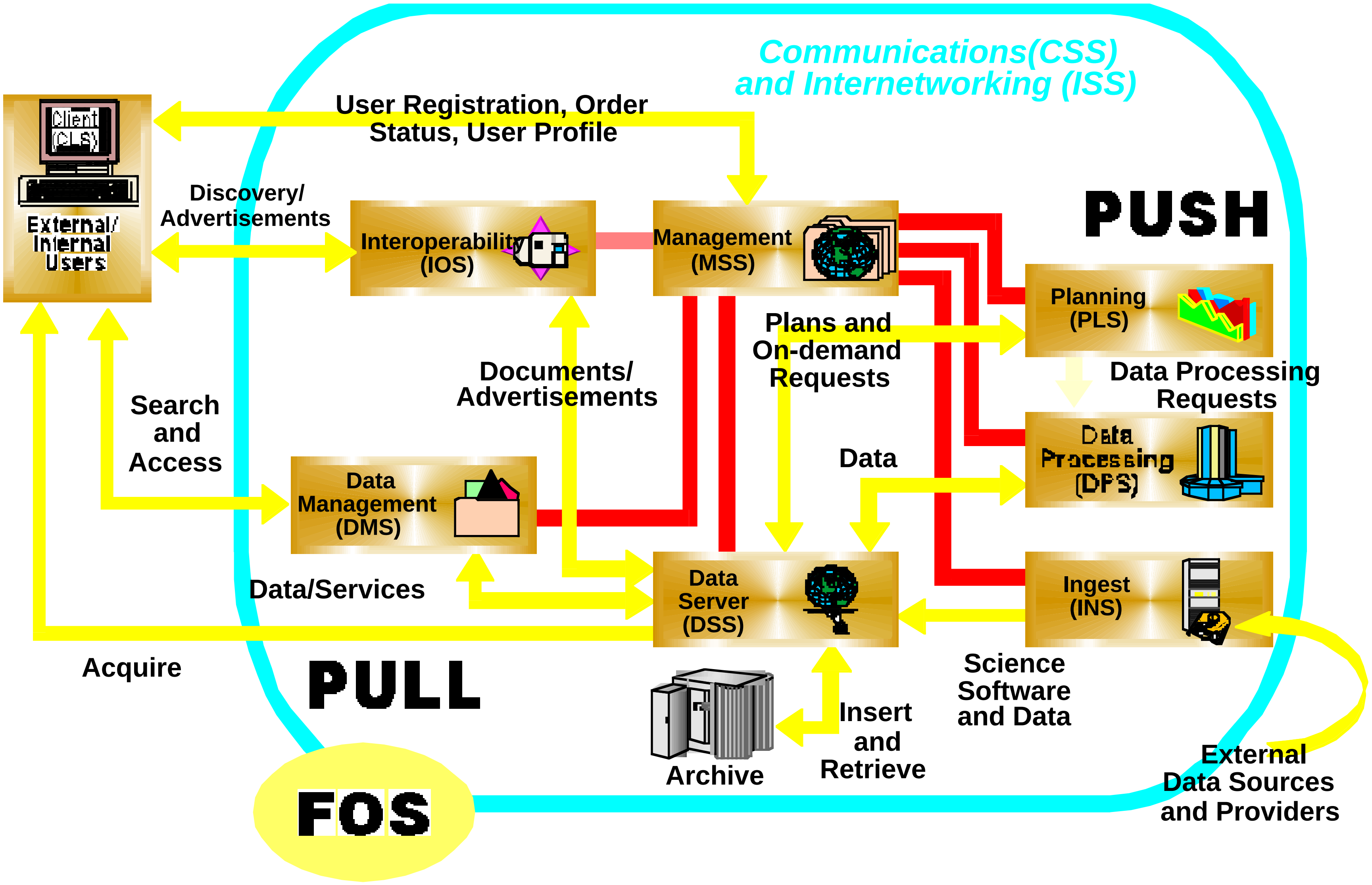


Figure 2.1-1. ECS DAAC Subsystems and Data Flows

MANAGEMENT—Provides functions for system startup/shutdown, resource management, performance monitoring, error logging, system and science software configuration management, and resource accounting.

COMMUNICATIONS and INTERNETWORKING—Provides the distributed computing infrastructure that enables intra- and inter-site communications between the subsystems.

FLIGHT OPERATIONS SEGMENT (FOS)—Provides the operations center for the U.S. EOS spacecraft and the U.S. EOS instruments, and coordinates mission operations for other non-U.S. EOS instruments on-board the U.S. spacecraft. Performance of the FOS is outside the scope of this Plan.

2.1.2 Potential Performance Bottlenecks

When potential users of ECS think of whether the system will meet their *performance* needs, they may ask one or more of the following key questions:

- Can each DAAC's science processors keep up with the processing load from the product generation executives (PGEs)? Issue: Throughput of the Data Processing Subsystem (DPS).
- Does each DAAC have enough bandwidth to move the required huge amounts of data into, within, and out of ECS in a timely fashion? Issue: Data transfer throughput.
- Can the DAAC infrastructure keep up with all of the needed low-level traffic (logging, remote procedure calls (RPCs), etc.)? Issue: Communications Subsystem (CSS) and Internetworking Subsystem (ISS) throughput.
- Can the archives (tape and cache) support all the required concurrent traffic (ingesting, staging, and destaging)? Issue: Ingest Subsystem (INS) and Data Server Subsystem (DSS) (especially archive) throughput.
- Can each DAAC support the automated planning for and execution of thousands of PGEs per day? How responsive is the system if dynamic replanning is required? Issue: Planning Subsystem (PLS) throughput.
- Can I search for, find, and get the data I want out of ECS in a timely fashion (using the medium I prefer)? Issue: DSS, Systems Management Subsystem (MSS), Interoperability Subsystem (IOS), and Data Management Subsystem (DMS) throughput.

2.1.3 ECS Performance Requirements, Workloads, and Goals

ECS has been designed to meet all of the expected simultaneous workloads. The formal requirements for the system come from NASA's *ECS Functional and Performance Requirements Specification* (F&PRS) and from the Technical Baseline for the ECS Project (described below in Appendix A.2.1). These requirements correspond to a variety of "push" (ingest, production processing, and archive) and "pull" (end-user search, subset, science product ordering and distribution) workloads on the system, and have been analyzed to produce expected utilizations of each major component of the ECS design. These expected utilizations were the basis for the hardware design and sizing of ECS. Figure 2.1-2 shows the process for deriving ECS performance requirements from the Technical Baseline and F&PRS documents.

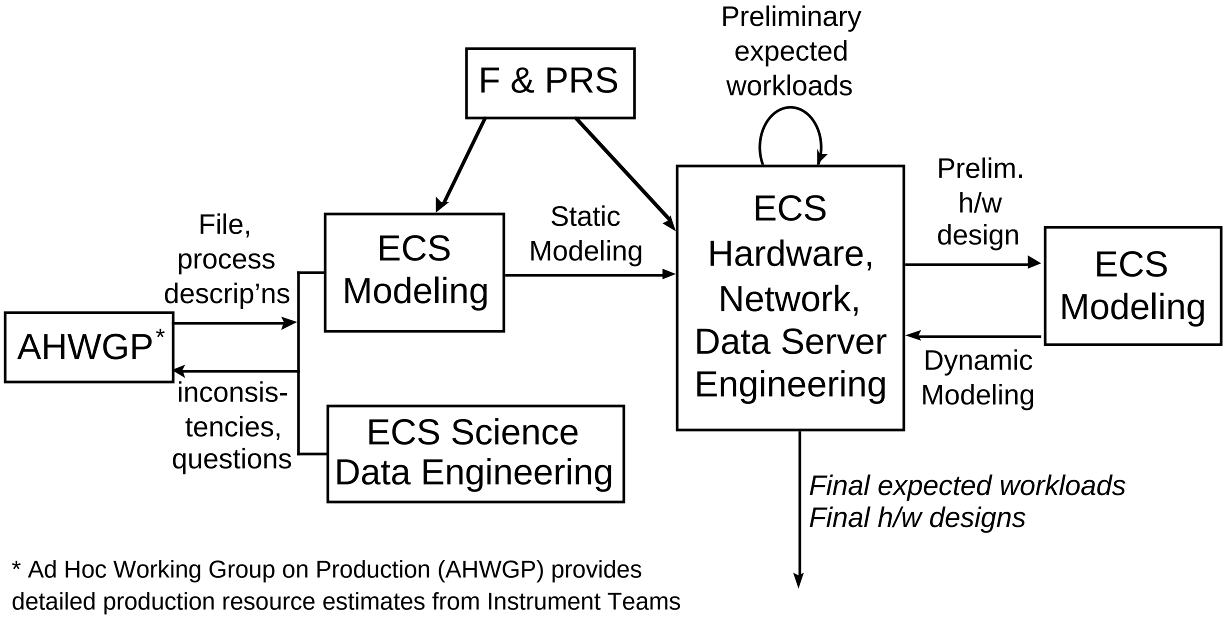

F & PRS
Preliminary expected workloads
ECS Modeling
Static Modeling
ECS Hardware, Network, Data Server Engineering
Prelim. h/w design
ECS Modeling
Dynamic Modeling
File, process descrip'ns
AHWGP*
inconsis- tencies, questions
ECS Science Data Engineering
Final expected workloads
Final h/w designs
* Ad Hoc Working Group on Production (AHWGP) provides detailed production resource estimates from Instrument Teams

Figure 2.1-2. Process for Deriving ECS Performance Requirements

Tables 2.1-1 shows, for each DAAC, the expected utilization and the design goals for the major ECS components. Table 2.1-2 is a version of Table 7-4 of the F&PRS, updated to correspond to the current ECS architecture.

Table 2.1-1. DAAC-by-DAAC Subsystem Workloads and Performance Goals

Potential Bottleneck	DAAC	Expected Utilization (@Launch)	Design Figure (@Launch)	Expected Utilization (@Launch + 1 yr)	Design Figure (@Launch + 1 yr)
Throughput rate from Source to Ingest to Archive (MB/sec)	EDC	3.29	8.0	3.29	8.0
	GSFC	1.11	8.0	1.11	8.0
	LaRC	0.57	8.0	0.57	8.0
	NSIDC	small	8.0	small	8.0
Throughput rate from Archive to Production (MB/sec)	EDC	7.1	40.0	14.2	40.0
	GSFC	6.4	35.0	12.8	70.0
	LaRC	2.7	15.0	5.3	25.0
	NSIDC	0.4	8.0	0.7	11.0
Throughput rate from Production to Archive (MB/sec)	EDC	7.2	20.0	14.4	20.0
	GSFC	5.4	26.0	10.8	52.0
	LaRC	2.7	10.0	5.3	20.0
	NSIDC	0.4	5.0	0.7	8.0
Throughput rate from Archive to Distribution (MB/sec) (total electronic + media)	EDC	2.7	30.0	3.2	30.0
	GSFC	8.3	35.0	15.4	70.0
	LaRC	3.4	15.0	6.1	30.0
	NSIDC	0.3	5.0	0.5	9.0
Number of MFLOPS actually delivered by the science processing hardware (steady state)	EDC	2,731	4,950	3,251	5,500
	GSFC	5,853	8,800	10,731	11,075
	LaRC	8,134	12,650	15,062	15,625
	NSIDC	53	274	98	274

Table 2.1-2. ECS-Wide User Workload Characteristics and Performance Goals

Session Category	Number of IMS Operations per Hour	Specific Operation	Response Time Requirement (secs)	Response Time Design Goal (secs)
Log-on and Authorization	100	Account confirmation and authorization	13	6
Directory Search	80	Search by single keyword attribute	8	2
		Search by multiple keyword and time or space range check	13	7
Inventory Search	120	Search one instrument by multiple keyword attribute w/time or space range check (one DAAC)	8	2
		Search multiple instruments by multiple keyword attribute w/time or space range check (one DAAC)	18	7
		Multiple DAAC inventory search by keyword attributes and time and/or space range check	58	11
Status Check (account or request)	60	Status of pending order or Data Acquisition Request	13	10
		Account status retrieval	13	6
Browse (for data selection)	50	Retrieve and begin to display standard pre-computed browse product	58	49
Ordering Services	25	Local DAAC order submission and confirmation	13	12
		Remote DAAC order submission and confirmation	38	30
		Order cost estimate	13	12

The remainder of chapter 2 describes the methodology and activities related to measuring and tuning ECS performance to cover the requirements and answer the questions listed above.

2.2 Performance Measurement and Tuning Methodology

2.2.1 General

The effort to characterize and tune ECS's performance will employ a set of performance scenarios to drive the system while performance data is collected. Each scenario is a processing thread that exercises selected functionality. The collection of thread-level scenarios will be aggregated into a system-level scenario that is representative of full DAAC operations.

The threads to be measured and tuned are discussed below in section 2.3.1. Each thread will be run in an instrumented environment that simultaneously captures performance-related data from the ECS custom software and system performance/utilization statistics (for central processing unit (CPU), disk, network, etc.) for each processor participating in the thread. The instrumentation data from the ECS processes will be analyzed to determine the rate of data movement through the system and the timing of process control events (e.g. client call to server process, server response, etc.). The system performance data will be analyzed to provide a time profile of the consumption of system resources. Analysis of this data will focus on determining how well the thread is performing compared to the benchmark performance of the system components, identifying those places in the design where parameters may be adjusted to yield performance improvement, and adjusting system parameters so that the system meets or exceeds its performance requirements/goals.

A template has been developed to assist in planning and execution of performance scenarios. The template appears in Appendix B. An example of a filled-in template appears in Appendix C.

2.2.2 Software Performance Logging

Data movement and timing information from the system will be collected from several sources described in this section.

2.2.2.1 Performance Logging

Log calls strategically placed in the code are the most effective method for collecting software performance information. For this reason a performance logging object was developed for use in this effort. The object type is *EcUtPerfData*, and its use is described in detail in the document "Performance Logging Guidelines" (June 24, 1997) and in a set of subsystem-specific supplements. To determine the appropriate points in the software to place the log calls and what data is to be logged, the performance team conducted meetings with the technical leads from each CSCI. The versions of ECS software to be used prior to the August Demo include performance logging calls within Science Data Server (a component of Data Server) only. The coding of performance instrumentation for Storage Management software is in progress and will be included in the baseline just after the August Demo. Ingest will not require new performance logging since it already records equivalent data in two performance data tables in its database (see next paragraph).

2.2.2.2 DBMS Logs

Ingest has two related tables used by its processes to record performance related data. Data in the tables includes start/completion times for Ingest transactions and the amount of data processed in the transaction. This data will be retrieved from the database management system (DBMS) after any performance thread involving Ingest has completed.

2.2.2.3 Screen Captures

ECS server processes can be configured to send status information to the console window from which the server is started. Depending on the verbosity of the output, there may be useful data in these screen messages. Therefore the performance team has developed and tested a script written in Perl which can be used to run the server start scripts. When this is done, each entry written to the standard output in the console window is preceded by a time stamp, "[hh:mm:ss]". In addition a copy of the time-stamped output is automatically written to a log file. This mechanism for data capture has been used successfully on the RC-2 version of the system.

2.2.2.4 Directory list

A major function of the system involves the movement of large files. A simple mechanism to monitor this movement is to perform periodic directory listings (UNIX `ls -ls` command) as files are written to the directory. While there is a limit on how small the time between samples may be set, it is possible to select a reasonable value where file transfers are large. For example, many large files will require several minutes to complete, so that a 5 second sleep between calls to `ls -ls` provides a reasonable set of samples.

2.2.3 System Performance Measurement

Several system utilities will be used to collect resource consumption information during the conduct of a performance thread. These include:

`sar`	provides CPU utilization
`sar -d`	provides disk utilization
`nfsstat`	provides client and server statistics from Network File System (NFS) processes
`sysperf`	provides AMASS system utilization

The first three sets of data are gathered using a set of Perl scripts run on each processor involved in the thread. The scripts parse the data as it is gathered and put it in a tabular form ready for import into Excel (spreadsheet). The `sysperf` program has a screen-oriented display that cannot be directed to a file or parsing script in the normal fashion. Therefore this data is gathered using the `script` command to initiate terminal logging followed by execution of the `sysperf` command. The resulting log file is post-processed to put it into spreadsheet import format.

Finally a network analyzer will be placed on the system to monitor fiber distributed data interface (FDDI) network traffic. The analyzer will employ an add-on that is designed to follow Sybase DBMS activity to and from key DBMS servers.

2.2.4 System Tuning

The data described in the previous section should provide sufficient data to allow the performance team to identify those components whose tuning will reap the greatest benefit to the performance of the system. These components include:

- Redundant Array of Inexpensive Disks (RAID) Subsystem
- AMASS® Tape systems and cache
- Network File System (NFS) configuration
- Sybase® servers
- ECS custom software (the change of last resort)

The details of which components and parameters are likely targets for tuning will be discovered as a result of executing this Plan. As parts of the system are identified that may require tuning, appropriate specialists will be consulted to identify and implement configuration changes.

Scenarios will be re-run to determine the effect of the changes and the entire process repeated until the final system requirements are achieved.

2.3 Performance-Related Activities

This section contains a discussion of the major performance-related activities to be accomplished between now and the Landsat-7 and AM-1 launches. A master schedule of these activities appears in chapter 3.

2.3.1 Thread Testing

The methodology for performance thread testing is described above, in section 2.2.

Performance threads to be tested include:

1. Determine the throughput rate from Source to Ingest to Archive.
2. Determine the throughput rate from Archive to Production.
3. Determine the throughput rate from Production to Archive.
4. Determine the throughput rate from Archive to Distribution.
5. Benchmark the performance of a PGE executing with the science computing facility (SCF) version of the Science Data Processing (SDP) toolkit and executing in the DAAC environment with the DAAC version of the toolkit. [The following performance data will be collected for each PGE: number of block input operations, number of block output operations, number of page faults, number of swaps, and cpu time.]
6. Determine the number of MFLOPS actually delivered by the science processing hardware.
7. Evaluate the effects of Data Server DBMS insert & acquire contention.
8. Evaluate the effects of AMASS® insert & acquire contention.
9. Evaluate the overhead of Communication Subsystem (CSS) message handling.
10. Determine Science Data Server query response times.
11. Determine user interface response times.
12. Evaluate the performance of the Planning database for a 30-day plan.
13. Determine the throughput rate from Data Distribution to hard media.
14. Evaluate Subscription Server response times.
15. Determine error & event logging overhead.

Thread number one has been tested and we report its results in Appendix C, Ingest to Archive Performance Scenario Plan.

2.3.2 Endurance and Stress Testing

ECS recognizes the importance of performing endurance and stress testing of ECS as part of the operational test and evaluation of the EOSDIS Ground System (EGS). This type of testing—usually characterized by running the system end-to-end for an extended period of time, and by introducing loads that exceed design requirements—is useful both in fine-tuning operational procedures and in identifying load and stress related problems. Typical performance problems surfaced in such tests are poor performance when caches are overrun, failures when queues exceed their expected maximum lengths, and degraded database performance as transaction rates increase and databases fill to operational sizes.

ECS will participate with ESDIS to plan a 72 hour end-to-end operational test of ECS during the EGS test and integration period. ESDIS will act as the executive agency for this plan, and will determine the calendar period during which this test can be performed. The goal will be to exercise all ECS code and functionality simultaneously at all Release B.0 sites. Ideally, this will include complete integration of the B.0 science code, and integration with all ECS external interfaces.

Such an undertaking is an enormous task both to organize and to perform and will take the concerted efforts of ECS, ESDIS, the Instrument Teams, the DAACs, EDOS, EBnet, NASA Integrated Services Network (NISN), Landsat Processing System (LPS), and other external data providers. A 72 hour test requires massive test data preparation. However, such a test will provide the greatest possible confidence that EGS is ready for operations.

If this level of integrated testing is not possible, the first fallback will be to perform an integrated test of ECS integrated with the launch versions of science software, simulating external interfaces such as EDOS. This will still provide most of the benefits of endurance and stress testing for ECS, but will not provide verification of the external interfaces.

2.3.3 Component-Level Measurement and Tuning

ECS is a complex system and its overall performance—from the perspectives of throughput, response time, and reliability—will be determined by complex interactions between its many components. However, the performance of some of these components can be measured and tuned comparatively directly and easily; the behavior of these components is generally dominated by the behavior of Commercial-Off-The-Shelf (COTS) hardware and software. Significant component-level measurement and tuning is planned for the areas of data transfer performance, database performance, and archive performance. Data transfer performance is further subdivided into three measurable areas: magnetic disk throughput performance, network throughput performance, and system throughput performance.

2.3.3.1 Magnetic Disk Performance Testing

The large data transfer rates between Ingest and the data archive, between the data archive and Processing, and between the data archive and the Distribution devices represent the single most difficult performance challenge of the ECS project. Magnetic disk performance—the rate at which hosts in the ECS system can read and write to their local disks—is a critical factor in these data transfers. ECS sustained average transfer rates rapidly overwhelm the memory buffers used by operating systems, hence the transfers cannot occur any faster than the sender can read from disk, or the receiver can write to disk.

During the system design phase, performance benchmarking was performed on various types of magnetic disk hardware in order to establish performance design points for system sizing. These design points—usually expressed as megabytes per second transferred per unit of equipment, under a defined load—were used to determine how much of what type of magnetic disk was procured for each data-intensive subsystem at each site.

These disk subsystems have now been installed and configured. As part of the Performance Assurance Plan, the Hardware Engineering organization is executing benchmarking tests against the hardware to measure disk throughput. These tests will be performed on the data storage areas used in those hardware configuration items (HWCIs) that manage the system data flows:

1. Ingest client hardware (ICLHW);
2. Data Repository hardware (DRPHW)
    a. Storage Management Cache, and
    b. AMASS Cache;
3. Science Processing hardware (SPRHW);
4. Access Control and Management hardware (ACMHW) [Electronic Distribution Cache];
5. Working Storage hardware (WKSHW) [Interim Files]; and
6. Distribution and Ingest peripherals hardware (DIPHW) [Media Distribution].

Each of these HWCIs has one or more large disk pools that must meet a required throughput. The performance of each of these disk pools will be tested.

The test approach begins by reviewing the installation configuration to make sure that the configuration conforms to the design, including the best known values for the tuning parameters offered by the device. Note that of the disk pools identified above, all are hosted on SGI RAID arrays except the Data Distribution cache, which is hosted on a Sun SPARC Storage Array; hence the number of types of equipment being tested and tuned is limited. If the initial configuration is not correct, the configuration is corrected/tuned before the testing starts.

The test tool used for these measurements is an instrumented version of the UNIX `dd` command called `lmdd`. `lmdd` allows the movement of data from memory to disk, from disk to memory, or from disk to disk. It captures performance statistics for these transfers.

On the SGI, `lmdd` allows testing using both filesystem cached I/O and Direct I/O. UNIX filesystems normally cache disk I/O in memory buffers; this caching can improve performance if the data transfers are being requested in very small blocks, or if certain data are accessed repeatedly. This caching is typically done in block sizes of 4 KB, 16 KB, or 64 KB, depending on the vendor and the version of the operating system. The Direct I/O `lmdd` option allows these buffers to be bypassed. This reduces the overhead associated with each transaction, and makes it possible to access the disk devices using much larger block sizes. Because each transaction to a disk drive has a huge overhead—including moving the read/write heads—disk performance for small blocks (64 KB and less) is dominated by the overhead time, and performance is poor. For that reason, ECS on the SGI makes use of Direct I/O and large block sizes (128 KB and up) wherever possible. Tests are run using various file sizes (from 1 MB to 1 GB) and Direct I/O tests are run using various block sizes (128 KB to 1 MB).

The results of the tests will be compared to the required throughput for the storage pool. If the results do not exceed the requirements, the configuration will be reviewed, tuned if necessary, and re-tested; if the hardware still does not meet the requirements, alternative system configurations (including the addition of hardware) will be investigated to meet the requirements.

The typical ECS SGI RAID configuration consists of one or more groups of five disks, formatted for RAID 3. Associated with this set of disks is a RAID controller and a SCSI-2 fast wide differential path to the host. A RAID controller may control from one to four disk groups, but typically only one or two, as ECS disk systems are actually more throughput-driven than volume-driven. The SGI xfs filesystem utility is used to stripe a filesystem across the pool of SCSI channels. For high performance subsystems, the total configuration may include eight SCSI-2 channels and eight controllers, with eight or sixteen disk groups. This configuration has been benchmarked at up to 75 MB/sec for reads. Performance of this configuration can be improved by providing more SCSI-2 channels and RAID controllers, or in some cases by providing more disk groups.

The Data Distribution SPARC Storage Array configuration consists of a fibre channel path from the host to the Storage Array. The Storage Array has a set of internal SCSI channels. The Storage Array is configured to perform RAID 0,1—to use mirror disks, with striping across the multiple internal SCSI channels. Performance of this configuration can be improved by increasing the number of disks (spindles) on each internal SCSI channel, and by tuning the stripe block sizes to the application.

This testing has been performed for the Ingest, Data Repository, Science Processing, and Data Distribution hosts of the Mini-DAAC, in preparation for the August Demo. These machines were tuned to meet the performance requirements for the Demo. A subset of these results is provided in Appendix A.

Preliminary testing has been performed at GSFC; however, important configuration changes (lessons learned) must be applied to GSFC equipment before additional testing is worthwhile. In particular, the SGI RAIDs were originally installed using RAID 5; however, the tuning efforts have shown clearly superior performance with RAID 3. The GSFC SGI RAIDs will be re-built accordingly. The Sun Storage Arrays were originally installed using RAID 5, but because the Storage Array is a software RAID—all RAID calculations are performed by the host—the performance in this configuration is unacceptable; instead, the arrays are being configured using RAID 0,1. This change also must be made at GSFC.

These changes will be performed at GSFC before September 5th, allowing baseline disk performance measurements to be performed by September 12th.

Tests at EROS Data Center (EDC), Langley Research Center (LaRC), and National Snow and Ice Data Center (NSIDC) will be performed remotely from Landover. The first step of this testing—checkout of the configurations—has begun; plans for any configuration changes (application of lessons learned) will be completed by August 29th, and the required changes will be completed by September 19th. This will allow the collection of baseline performance measurements by September 26th.

Remedial actions and re-testing at each site will be scheduled as necessary.

2.3.3.2 Network Performance Testing

The second potential bottleneck in transferring data from point to point within the ECS system is the ECS internal network at the DAAC. The ECS internal network for each DAAC was designed based upon the expected data flows from host to host at the DAAC, determined from the system data flow analysis (push and pull requirements) applied to the hardware and software architecture.

Performance benchmarking was also performed during the design phase on FDDI and high performance parallel interface (HiPPI) network equipment in order to establish performance design points for network sizing. These design points were usually expressed in megabytes per second throughput, and were used to design the load-handling capacity of the ECS internal networks.

The Release B networks have been installed at each of the sites. As part of the Performance Assurance Plan, the Network Engineering organization is executing benchmarking tests against the networks to measure network throughput. These tests will be performed between the same platforms identified above for the magnetic disk throughput performance tests.

The test approach begins by reviewing the ECS DAAC network to make sure that the configuration conforms to the design, including the best known values for the tuning parameters offered by the network devices. If the initial configuration is not correct, the configuration is corrected/tuned before the testing starts. The configuration items of importance include the attachment of the correct hosts to the HiPPI network and to the correct FDDI concentrators or FDDI switch ports.

The test tool used for these measurements produces a stream of TCP/IP packets that are sent from one host to another. These packets are created in memory by the test software on the sender, and are discarded by the receiver; hence, these tests are independent of the magnetic disk performance on the sender and receiver. The test tool allows the size and number of packets to be varied, and collects performance statistics for the transfers.

Testing will be performed for both the FDDI network and the HiPPI network, at sites having a HiPPI network (GSFC, LaRC, and EDC).

The results of the tests will be compared to the design expectations. FDDI connections of this type are expected to provide 8 MB/sec; HiPPI connections are expected to provide 70 MB/sec. If the results do not exceed these levels, the configurations will be reviewed, tuned if necessary, and re-tested; if the hardware still does not meet the requirements, alternative system configurations will be investigated to meet the requirements.

This testing can be performed more or less in parallel with the magnetic disk performance testing discussed above, for all steps except actual data collection. (As in most benchmarking efforts, configuration checks and changes and test set-up are expected to consume 90% or more of the total allocated test time.) Because there have been no lessons learned to date that significantly affect the network configurations, it is believed that these test can proceed directly. Testing at GSFC is expected to complete by September 5, with completion at EDC, LaRC, and NSIDC by September 19th.

2.3.3.3 System Data Transfer Performance Testing

The actual data transfers from one host to another within ECS, with the exception of process to process messages, are accomplished using standard operating system file transfer mechanisms. The next logical step after testing the disk and network throughputs using low level drivers is therefore to test the ability of the system to transfer data from the disks of one host to the disks of another host using these mechanisms, which include ftp, dd via nfs, and dd via SGI's Bulk Data Service (bds).

The principal data transfers through the system and their mechanisms are identified as follows:

1. External network interface to Ingest via ftp over FDDI;
2. Ingest to AMASS cache via nfs over FDDI;
3. AMASS cache to D3 tape over SCSI-2 bus;
4. AMASS cache to Storage Management Cache via dd over SCSI-2 bus;
5. Storage Management Cache to Science Processor via ftp over HiPPI;
6. Science Processor to AMASS Cache via bds over HiPPI;
7. Working Storage to Science Processor via ftp over HiPPI;
8. Science Processing to Working Storage via bds over HiPPI;
9. Storage Management Cache to Data Distribution via nfs over FDDI; and
10. Storage Management Cache to Access Control and Management via bds over HiPPI.

Note that at NSIDC, Ingest functionality is hosted on the Access Control Management HWCI, and HiPPI is not implemented; transfers shown above as occurring over HiPPI occur over FDDI at NSIDC.

Note also that item 3 above, the transfer of data by AMASS between its disk cache and tape drives, is considered a separate performance measurement area, and is discussed below in section 2.3.3.4. The transfer of data between the AMASS cache and the Storage Management Cache, item 4 above, while not a host to host transfer, is included in the system data transfer tests because it places a load on these two disk pools.

Each of these transfer mechanisms will be tested. In cases where there is more than one server or processor in a HWCI (e.g., multiple Science Processors), each path will be tested. Tests will also be run with multiple instances of the transfers happening simultaneously—for example, multiple files being transferred from Ingest to the AMASS cache in parallel. These test results will be compared to the disk and network throughput results for the DAAC for self-consistency; anomalous results will be investigated. The results will also be compared to the requirements derived from the system analysis performed during the design stage; see Table 2.1-1 above. If performance shortfalls are identified, tuning efforts will be undertaken; if necessary, additional hardware will be added to the configuration.

Upon completing each of these point to point tests, tests will be performed at each site in which transfers will occur along all data paths simultaneously. These tests will be run first with one instance of each transfer along each path; they will then be run with multiple instances of each transfer happening in parallel. The test results will be examined for patterns in the total throughput, to understand the maximum throughput of the system under stress, and to observe any degradation in performance as the load increases. If necessary, tuning will be performed (and possibly hardware added) until the performance under this stress test meets the system requirements.

This testing has already been performed in the Mini-DAAC, to support the August Demo, on the Ingest Thread path—the external (virtual EDOS) server, the Ingest Server, and the Data Repository Server. Results of this testing are synopsized in Appendices A and C.

This testing is most usefully performed after the satisfactory completion of the magnetic disk performance testing and the network performance testing. Hence, this testing is expected to commence at GSFC on September 15 and complete September 26. At EDC, LaRC, and NSIDC, testing will commence on September 30 and complete October 10.

2.3.3.4 AMASS Tuning

The ability of the ECS archive to absorb and to serve data at required rates depends on a well integrated, well tuned combination of high performance archive hardware and software. All the ECS DAACs have the same architecture and constituent components. The DAACs differ only in the size and particulars of equipment.

AMASS is the File Storage Management System (FSMS) Commercial Off The Shelf (COTS) product that manages the physical aspects of the Science Data Server archive. To date, all of the tuning and configuration adjustments for improving the performance of the archive component were made against either the AMASS software itself, or the SGI RAID used as a temporary cache, for the data passing to and from the archival silo. Therefore, “AMASS Tuning” is used as a de facto synonym for the tuning of the overall archival component. AMASS is also the planned FSMS driver for the Browse Data component, which remains to be tuned for operation.

The following “AMASS Tuning” activities are planned between August 1997 and the scheduled launch.

2.3.3.4.1 STK®-Based Archive Performance Improvement Activities

1. RAID Performance. Sizable performance improvements have been realized as described in Appendix Section, A.3.2, AMASS Performance. During the “Tiger Team” capacity planning activity, in May of 1996, the peak overall throughput to an STK Powderhorn silo was assumed to be at 5 MB per drive. Therefore, the performance target for a silo equipped with eight drives would be a peak throughput of 40 MB/sec. The current read-from-tape performance, 29 MB/sec, is on the order of 70% of the target performance.

AMASS 4.9, expected from the vendor in September 1997, will allow for greater than 2 GB AMASS Cache partitions. At that point, the RAID portion of AMASS RAID will be configured to its planned capacity of approximately 0.25 TB. A series of tuning measures will be tested at that point to attempt to bring the peak performance closer to the target 40 MB/sec. If the throughput with that configuration remains inadequate, an alternative would be to test the use of a third party supplied Fibre Channel attached RAID, that would allow for higher data rates.

2. Continued AMASS corrections by the vendor intended to improve performance. Test the AMASS corrections promised by the vendor: asynchronous mount, elimination of the tape drive hardware buffer flush (savings of up to 20 seconds per four cache blocks).

3. Further performance testing of the archive configuration using ECS Storage Management code for caching, with test scenarios and test file sizes as predicted for normal post-launch archive operation.

4. Establish optimal partitioning of the archive data into volume groups. Volume groups determine the physical associations among data product granules within the tape archive(s).

5. As part of functionality testing, test backup/restore and failure recovery scenarios.

6. End-to-end DAAC testing for the purpose of end-to-end system performance improvement.

2.3.3.4.2 EMASS AML-Based Archive Performance Improvement Activities

1. Configure and tune the Browse Data archive stored on the optical media. Configuration and tuning of the optical media based archive is similar to that of the STK based archive. Lessons learned during performance tuning of the STK based configuration, as described in Appendix Section A.3.2, AMASS Performance, will be used in this exercise. The peculiarity of tuning for the Browse Data lies in the very small, 1 MB, file sizes and a quick random access pattern to the optically stored data, unlike the sequential access to the data stored on tape media.

Configurable components: RAID, AMASS cache

Tunable parameters: volume group allocation, if any; blocking factors

2. Configure and tune AML-based NTP archive. Configuration and tuning of the NTP based archive is similar to that of the STK based archive. Lessons learned during performance tuning of the STK based configuration, as described in Appendix A.3.2, will be used in this exercise.

2.3.3.5 Database Tuning

Databases managed by the Sybase Relational Database Management System (RDBMS) are ubiquitous throughout ECS. Most of these databases are directly used by ECS custom code; others are used by ECS COTS software, notably Autosys and SmartStream. Each of these databases has an expected performance profile, including initial size, growth rates, transaction rates, and response times.

Analysis of database performance generally takes a multi-tiered approach. At the lowest level, the data access patterns are analyzed and the database physical design is tailored for these access patterns. These activities occur during the design stage of the development. During the implementation phase, actual transactions are inspected to make sure that they are efficiently coded, and are consistent with the expected use of the database. Once implemented, the performance of the database is measured under various loads; performance is tuned by modifying various configuration parameters provided by the RDBMS, and in some cases by modifying the database design or the data access code.

ECS Database Engineering is currently using Sybase tools to capture and evaluate the actual Structured Query Language (SQL) being executed by various ECS databases. The capture tool can be used to examine SQL being created directly by ECS custom code, as well as SQL created by API tools (such as RogueWave dbtools.h++) and COTS (Autosys and SmartStream). For the code created by the API tools, this methodology provides insight into the efficiency of the generated SQL; if this code is found to be unacceptably inefficient, the API calls can be altered to try for more efficient generated code, or the underlying database design can be altered to better accommodate the inefficient code. This monitoring also identifies how indexes are being used by the Sybase query optimizer, pointing out the need (or lack thereof) for particular indexes. This monitoring will continue through the completion of development and integration of database software for Release B ECS.

Once a sufficient base of appropriately sized databases, transactions, and test data have been developed, actual database performance can be assessed and tuned. Extensive testing and tuning has already been performed for the Science Data Server database as part of the Spatial Query Server and Illustra evaluation. This type of activity will be applied, on a smaller scale, to the other databases within ECS.

Actual performance can be assessed from a response time perspective or a capacity perspective. Response times can be measured in a variety of ways. For some of the system databases, a direct user interface is available that will provide a basis for assessing response time performance—for example, searches provided through the client. For these cases, the analytical task becomes the separation of the database component of the response time from all other contributors.

Other system databases do not have a direct user interface because they serve system internal functions; an example is the Subscription database. Response time performance for these databases may be measured using test drivers—as was done in the SQS and Illustra testing—or by capturing timing marks from instrumented code as test threads are run through the system.

Database capacity—usually measured as the number of transactions that the database can handle per second—is usually tested with test drivers, although it can also be measured when the system is put through thread tests, operational tests, or stress tests. Sybase monitoring tools can be used to measure the throughput of the system during such tests, although they add their own overhead and reduce the maximum capacity achievable. The Sybase monitoring tool collects an enormous wealth of information about the use of Sybase resources—everything from hardware (CPUs, memory, disk channels) to Sybase internally configured resources (cache blocks, spin locks, indexes). These resources can then be tuned according to the usage patterns observed (including the possible addition of hardware).

Database capacity performance monitoring will be piggybacked on other system level ECS testing, particularly the thread testing identified in section 2.3.1. Capacity monitoring will also be an important part of the Endurance and Stress Testing described in section 2.3.2.

2.3.4 Other Performance Risk Mitigation Activities

There are a number of ECS activities that, though not directly related to performance measurement, will contribute to the mitigation of ECS performance risks.

2.3.4.1 COTS Software Performance Survey

ECS System Engineering has undertaken an effort to reduce the risk associated with the performance of COTS software products by learning as much as possible from other end users of these products. The lessons learned data will be collected by conducting a survey of end users of the COTS software products that are considered to present the greatest performance risk to ECS.

In the first step of this effort, ECS engineers and architects were asked to rank the COTS software products used by ECS according to the performance risk the products present to ECS success. The top nine products on the ranked list (from greatest risk to least risk) were as follows:

1. Autosys
2. OSF's Distributed Computing Environment (DCE)
3. Object Oriented DCE (OODCE)
4. Sybase SQL Server
5. Sybase Replication Server
6. RogueWave DBTools.h++
7. BDS
8. SQS Server
9. AMASS

Two of these products were removed from the scope of the survey. Sybase SQL Survey was removed from the survey because of the depth of ECS program knowledge of the product and because of the large amount of material (including training) on Sybase performance available from the vendor. BDS was removed from the survey because of its newness—we expected to find very few users with the software in production environments—and because of the extensive, successful benchmarking that ECS has performed with the product.

The next steps in the survey effort are underway: identification of end users of these products, and preparation of the survey instrument. End users to be surveyed are being solicited from the product vendors, from ECS sources, and from EDS corporate sources. A general survey for all end users has been developed, and survey questions specific to each product are being written. As the surveys are completed and the end users are identified, the surveys will be sent to the users. When the survey results are received they will be tabulated, and follow-up phone interviews will be initiated to solicit further details. The survey will be completed in October, 1997.

The goal in performing the surveys is to elicit as much detailed information as possible concerning performance, resource, and configuration issues. ECS hopes to avoid any problems realized by other users of the products by proactively applying their lessons learned.

2.3.4.2 System And Requirements Analysis

ECS and its requirements continue to evolve. The data transfer requirements for ECS presented in Table 2.1-1 are derived from modeling and simulation of the ECS design and the ECS February 1996 Technical Baseline. The ECS Technical Baseline defines the data products to be produced and archived by ECS, and defines the Product Generation Executives (PGEs) that will produce the majority of the products. These inputs to the Technical Baseline come from the ECS Ad Hoc Working Group on Production (AHWGP), principally from the Instrument Teams writing the science software to be executed by ECS.

New inputs (since February 1996) have been received by the AHWGP for most of the AM-1 instruments. ESDIS and ECS are analyzing these inputs in preparation for an update to the ECS Technical Baseline. These changes in requirements will directly affect the data transfer requirements for the system, although the analysis to date indicates that the changes are minor.

A much more significant change is also being evaluated. On August 7, 1997, ESDIS directed ECS to plan the second Release B procurement assuming that ECS Science Processing of Level 2 and above products would be implemented over a three year period after launch. Level 2 and above processing would be phased in starting at 25% during the year after launch, moving to 50% during the second year after launch, to 75% during the third year after launch, and reaching 100% starting three years after launch.

Clearly, this reduction in the production of Level 2 and above products will significantly reduce the performance requirements for ECS during the first three years after launch, and significantly reduce the performance risks of the system. However, ECS expects that the Instrument Teams, given these limits on production, will most likely implement changes in their processing plans that are more complex than simple across the board reductions in processing requirements and product sizes; they may choose to move some production from one DAAC to another to make more efficient use of resources, or they may postpone the production of some products until well after launch. These processing plan changes will ultimately be translated into inputs to the AHWGP, and will in turn be analyzed and incorporated into a new ECS Technical Baseline. This new Technical Baseline will be used as input to the ECS dynamic simulation, which is the best analysis tool available for determining actual data flows between hosts. The results of this modeling exercise will be used to adjust the values in Table 2.1-1 to reflect the changes in the requirements, and to re-target the Performance Engineering efforts. This change is too recent for ECS to forecast when new AHWGP inputs will be available for analysis.

2.3.4.3 The Second Release B Procurement

Procurement of the system capacity required to support Release B and its instruments is planned in four annual increments. The first and largest of these procurements has been completed and the equipment has been installed, providing the basic infrastructure for the GSFC, LaRC, EDC, and NSIDC DAACs. The second incremental procurement, planned for the fourth quarter of

1997, will increase the Science Processing and archive capacity at these DAACs to the levels required for operations through one year after launch of AM-1.

The second procurement and its corresponding installation provides an opportunity to correct performance deficiencies that are best fixed through the addition of hardware. ECS is in general reluctant to throw hardware at problems—the ECS COTS costs continue to be closely monitored for growth. However in some cases—when the requirements or the design have changed, or the design performance points can't be achieved—adding hardware can be the most cost-effective solution to the problem.

The program schedule limits the window during which performance data can be used to guide the second procurement. In order to have the at-launch capacity upgrade installed, integrated, and tested at the DAACs prior to an anticipated launch freeze, the procurements must be initiated in early October. This will allow only early performance plan results to feed the procurement. If results obtained later during performance testing indicate a hardware shortfall, a special procurement, installation, and acceptance test will have to be executed. The August 7th direction from ESDIS to gradually phase production capacity has significantly impacted the second Release B procurement. Although it is expected that this change will eventually result in a new Technical Baseline, ECS is currently proceeding with the second procurement with the phasing limits applied across the board to the current (February 1996) Technical Baseline.

3. Schedule

The following page shows the schedule for the major performance-related activities before Landsat-7 and AM-1 launches. In our experience to date, the preparation, execution, and documentation of each thread takes approximately eight working days. This figure should shrink during fall, 1997, when the system is more complete and the ECS software is more stable.

ID	Task Name
1	August Demo Preparation
2	September Rump Session
3	Acceptance Test
4	Operational Test and Evaluation
5	L-7, AM-1 Launch
6	**Component-Level Measurement and Tuning**
7	Data Transfers
8	RAID Tuning
9	AMASS Tuning
10	Database Tuning
11	**Thread Testing**
12	Source-to-Ingest-to-Archive
13	Source-to-Ingest-to-Archive Revisit at GSFC
14	Archive-to-Distribution
15	Archive-to-Production
16	Production-to-Archive
17	DSS Insert & acquire contention
18	AMASS Insert & acquire contention
19	Real PGE: SCF vs. DAAC toolkits
20	Synthetic PGE proc. rates
21	CSS Message Handling
22	Real PGE: attainable processing rates
23	SDSRV query response times
24	User interface response times
25	Planning-DB performance
26	DDIST to hard media
27	SBSRV response times
28	Error & event logging overhead
29	Revisit Source-to-Ingest-to-Archive
30	Other thread testing & revisits
31	**Endurance and Stress Testing**
32	**Performance Risk Mitigation Activities**
33	COTS Software Performance Survey
34	Second Release B Procurement
35	Modeling and Simulation

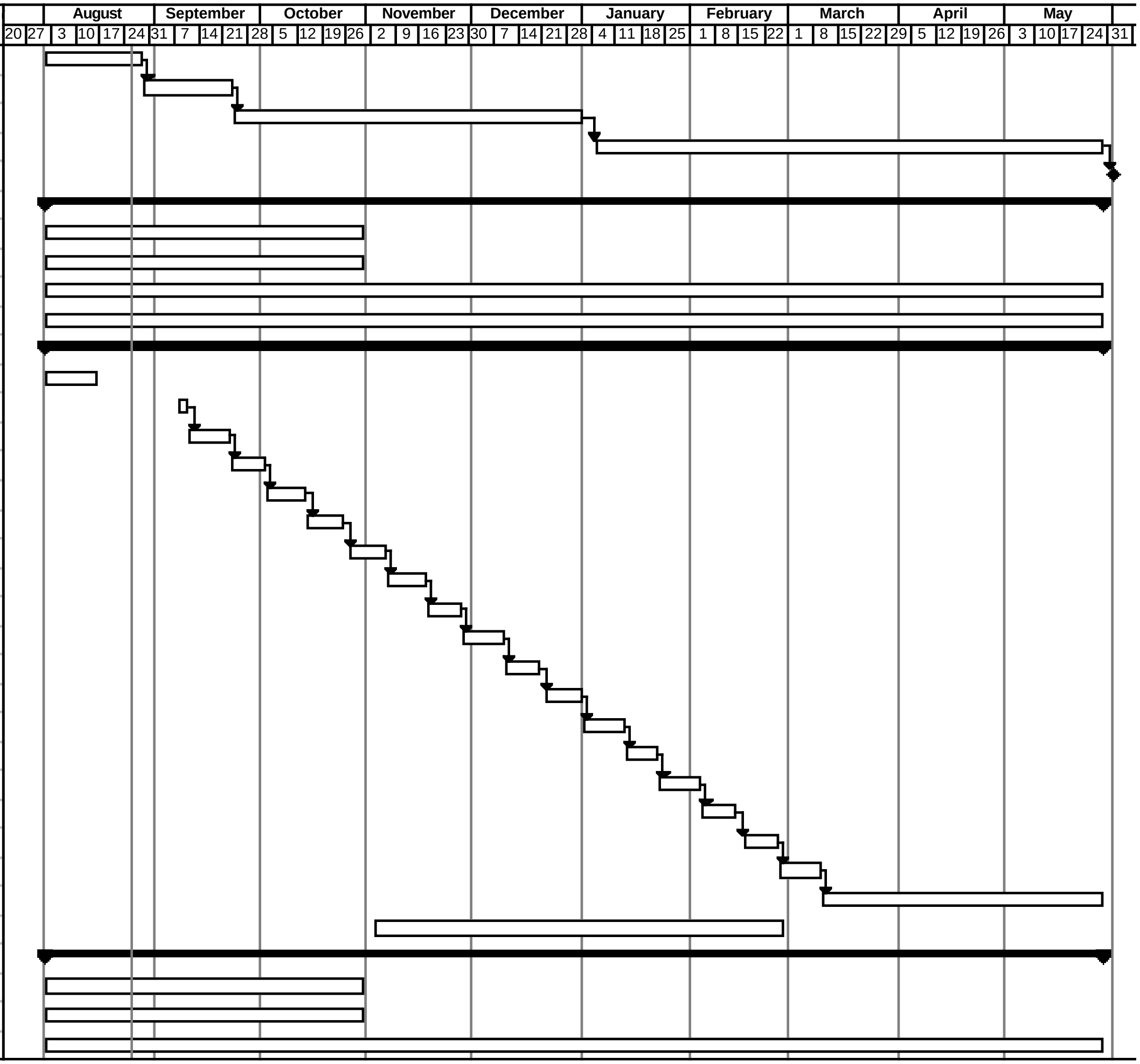


Figure 3-1. Schedule of Performance-Related Activities

Appendix A. ECS Performance Engineering Background

A.1 Introduction

The processes for ensuring adequate ECS performance have been operating for several years. The first major activity was and is system modeling. Even before there was an actual design, the ECS Modeling Group built a number of models to experiment with tradeoffs of architectural design, performance, and cost. As the design was developed and refined, these models have changed along with them, and so still provide predictions of system and even subsystem performance against various workloads. The second major activity was component benchmarking. ECS recognized early that certain components would be likely performance bottlenecks, and we wanted to learn as early as possible what performance could be expected and what improvements could be gained by tuning and reconfiguring. The third major activity was to form a Performance Engineering Tiger Team to monitor system performance and performance test plans, and to propose risk mitigation strategies. This team evolved into the Performance Measurement and Tuning Team, which is supporting the August (1997) Demo and will support future performance-related activities

A.2 Modeling

Even before the ECS contract was let, models have been used to predict how much hardware would be needed to keep up with the required workload. The following is a synopsis of the models used since the Preliminary Design Review (PDR)-A timeframe, and an indication of some of the results they produced. These models are documented in *Systems Performance Models*, 241-TP-001-001, June 1996 and were described in briefing format during January and February, 1996 in the ECS Modeling Workshops' documentation, 731-001-001 and 731-002-001.

A.2.1 Static Model

The static (spreadsheet) model is used to estimate input/output (I/O) and processor requirements for first-time push processing. The model is used to provide insight into the average and "busy day" magnitudes of the processing CPU and I/O loads on the Science Data Processing Segment (SDPS). This model is the first one executed for new push data and is the first step in sizing the SDPS.

All input data is from the Technical Baseline for the ECS Project: the operating hours by site and the Ad Hoc Working Group on Production (AHWGP) process description file for each site, instrument, and time period (epoch) that characterizes the push load on the system in terms of I/O volumes, PGE execution times, and frequency of invocation.

The process description file is sorted in order by epoch and instrument. The average number of million floating point operations per second (MFLOPS) is calculated for each PGE by multiplying the estimated number of millions of floating-point instructions required (MFLO) by the number of times per second that the PGE is executed. These requirements are then summed over the PGEs for each instrument and then summed over the instruments processed by a particular DAAC.

The average I/O bandwidth required for staging and destaging the data for each PGE is calculated. Results are accumulated for each instrument by site and by epoch. The same values are recalculated for the “busy day”. A busy day is when all PGEs with frequency of execution of less than once per day, are simulated to execute on the same day as the daily PGEs. This has the effect of simulating a day when all products need to be produced on the same day.

The results provide analyses of average and busy day and provide summaries for each instrument by epoch and by site for: the number of PGE invocations per day, the total MFLOPS required; and, the I/O bandwidth requirements (megabytes/second) for the local disk to processing, the host-attached backplane, and combinations of staging and destaging.

A.2.2 BONeS System Model

The System Performance Model was developed as a system simulation using the Block Oriented Network Simulation (BONeS)® tool. It is a dynamic, discrete-event simulation model which can support capacity planning, requirements analysis, design, and development of the ECS. The model includes the performance of external elements and is updated in parallel with the system development, simulating the as-developed system to allow performance checking of the completed system and evaluation of any changes proposed as modifications to the completed system. It is sufficiently detailed to permit it to be used to select and validate processor hardware and software architectures. It also can be used to simulate data flows from instruments to investigators, user interactions with the ECS or with individual instruments, and the processing workload resulting from these activities.

A.2.3 ECS End-to-End Model

The end-to-end analytic model is the primary tool for determining and analyzing the response times through all the subsystems in a DAAC. The model accounts for push, pull, infrastructure loads, and distribution of products. The model provides an average or steady-state view of the ECS.

The model input is a collection of threads partitioned into elements representing most of the work flows in the ECS and designed to account for nearly all work done in each subsystem. The thread elements include software executables and calls to other resources. Each thread and/or thread element activity has a multiplier corresponding to the frequency of invocation. and the quantity of resource used by each invocation. The input values for these activities are obtained from benchmarking, other models (e.g., the dynamic model) and from transaction estimates. The model will normally be run with the best combination of values available from any source for each activity.

The output from the model is: per site/subsystem/cluster—the average number of busy processors; the average number of read/write stations, and the percentage disk utilization; the LAN utilization by site; for each thread—the end-to-end execution time, the time profile (which activities occupy how much time) and throughput (activations/day), and pull workload response time vs. arrival rate. The results are also used as a validation of the results of other models.

A.3 Benchmarking

The following is a synopsis of ECS benchmarking activities through July 31, 1997:

A.3.1 Magnetic Disk Performance

The following tests of SGI RAID performance were performed in July, 1997 in the mini-DAAC on the machine lasher.

Activity: write a file to disk.

	Throughput (MB/sec)	
File Size	RAID L5	RAID L3
1 MB	71.92	66.38
10 MB	69.24	70.81
50 MB	66.88	61.68
100 MB	67.54	69.68
500 MB	2.56	12.29
1000 MB	2.53	8.87

Activity: read a file from disk.

	Throughput (MB/sec)	
File Size	RAID L5	RAID L3
1 MB	1.85	2.79
10 MB	7.87	14.03
50 MB	10.20	15.14
100 MB	7.10	14.03
500 MB	8.37	15.00
1000 MB	11.73	18.80

Comparison of RAID Level 3 vs. Level 5 performance:

	Throughput (MB/sec)			
Buffer size	128 KB	256 KB	512 KB	1024 KB
L5 Write	1.44	1.93	2.21	2.53
L5 Read	4.17	8.34	9.79	11.53
L3 Write	5.03	6.24	8.41	10.88
L3 Read	15.30	16.01	16.34	16.64

A.3.2 AMASS Performance

AMASS is the File Storage Management System (FSMS) Commercial Off The Shelf (COTS) product managing the physical storage for the Science Data Server. The stand-alone performance of the archive was assessed by measuring the data throughput to and from the tape drives in the robotic repository. The data was transferred locally. The only network transfers were of a very small volume of robotic control signals and physical inventory synchronization data between the AMASS software and the STK automated cartridge system library software (ACSLS) robotic control software.

The components requiring adjustment to improve data rates were the AMASS software and the SGI RAID disk attached to the AMASS host. This section summarizes the performance improvements and gives a high level description of the configuration changes that produced them.

A.3.2.1 Overall Configuration

Figure A.3.2-1, Archive Hardware and Software Configuration Under Test, depicts the overall configuration. For a representation of RAID attachment see Figures A3.2-2 and A.3.2-3.

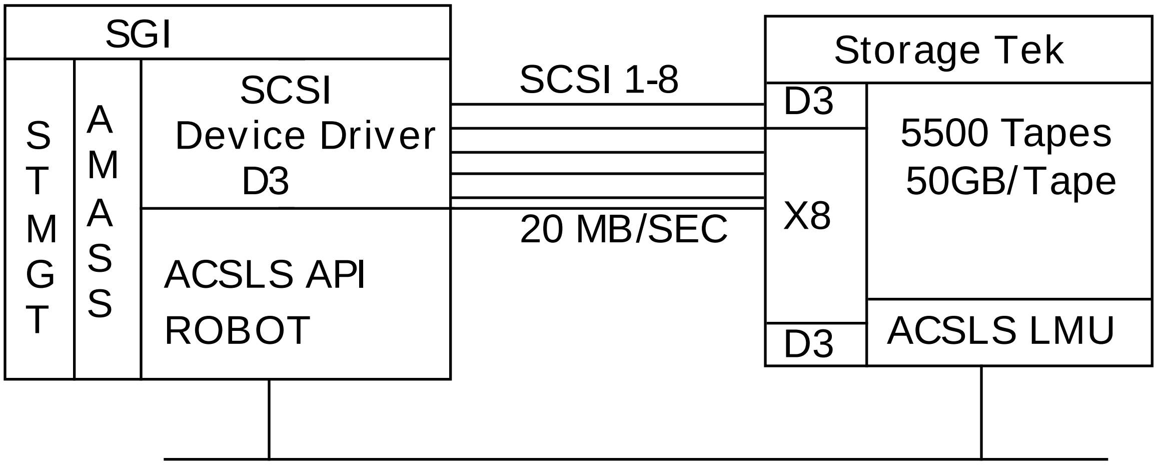


Figure A.3.2-1 Archive Hardware and Software Configuration Under Test

AMASS File Storage Management System (FSMS) software from EMASS Corporation controls the physical storage of the data in the repository and is hosted on a Silicon Graphics, Incorporated (SGI) multiprocessor Challenge class server. The data collection resides in the STK Powderhorn robotic silo and is recorded using D3 helical-scan tape drives from STK Corporation. SGI RAID disk is used for the temporary caching of data en route to and from the robotic silo.

As shown, the tape drives (D3) residing in the Storage Tek robotic silo are directly connected to the SGI Host via Fast-And-Wide SCSI II channels. Each channel is individually capable of the throughput of 20 MB/sec. Each of the eight tape drives is rated by the manufacturer capable of 11.2 MB/sec sustained throughput. As shown in table A3.3.1, the drives exhibited even higher streaming data throughput rates if hardware data compression was realized during recording. The compression feature was enabled during testing, but the degree of data compression realized in each case depended on the specific data used and in turn determined the data rate above the manufacturer's rating.

The control of the robotic mechanism of the silo (loading and unloading of the tapes) is effected via the STK ACSLS interface software running on a SPARC 5 SUN workstation. AMASS addresses the ACSLS through a network connection. The ACSLS controls the robot directly via an RS232 line.

A.3.2.2 Comparison of the Initial and Current Performance

Tables A3.2-1, Individual Tape Drive Throughput Improvements, and A3.2-2, Cumulative Archive Throughput Improvements, illustrate data rate performance changes as the result of the integration/tuning activity. The design goals were 5 MB/sec for each of the drives individually and 40 MB/sec for all eight drives when reading or writing simultaneously.

Table A3.2-1. Individual Tape Drive Throughput Improvements

	1/1997 Performance*	**8/1997 Performance***
Peak Write Rate	~ 2 MB/sec	16 MB/sec
Peak Read Rate	~ 2 MB/sec	16 MB/sec

***** Note: Compression is enabled.

Table A3.3.2. Cumulative Archive Throughput Improvements

	1/1997 Performance	**8/1997 Performance**
Peak Write Rate	7 MB/sec	33 MB/sec
Peak Read Rate	9.5 MB/sec	29 MB/sec

A.3.2.3 Summary of the Tuning and Configuration Adjustments Efforts

A.3.2.3.1 Hardware

The greatest degree of performance improvement was due to the enlarging and tuning of the RAID configuration. The maximum throughput rate of the disk configuration determines the ceiling on the peak cumulative throughput of the stand-alone archive configuration. The change in the RAID configuration level from 5 to 3 also accounted for the initial improvement in the individual throughput to and from tape. Most of the individual tape drive performance improvement is due to changes in the AMASS software that allowed for a configurable blocking factor on the tape as discussed further in this section.

Figure A3.2-2, RAID Configuration Under Test as of 1/1997, illustrates the configuration that was producing the initial throughput rates listed in the “1/1997 Performance” columns of tables A3.2-1 and -2. As shown, the RAID level configured initially was 5. Command tag queuing was not enabled. The available RAID disk was divided into 6 GB of raw disk for AMASS Cache and the remainder for user file system.

During a test of a write to tape, the data file is copied from a directory in the user file system partition to the raw AMASS Cache partition and subsequently to tape. In the read from tape test, the direction of data is reversed.

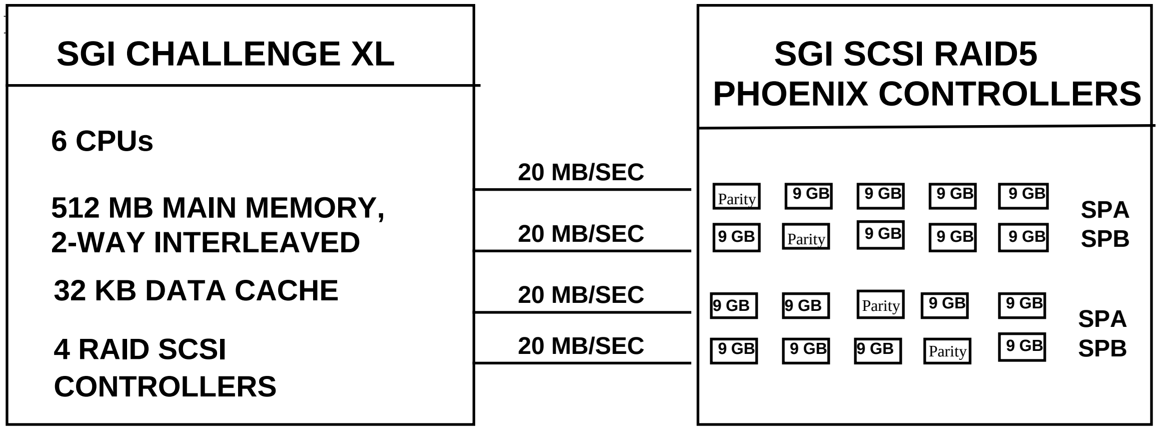


Figure A3.2-2 RAID Configuration Under Test as of 1/1997

An enlarged RAID configuration currently in use is shown in Figure A3.2-3, RAID Configuration Under Test as of 8/1997.

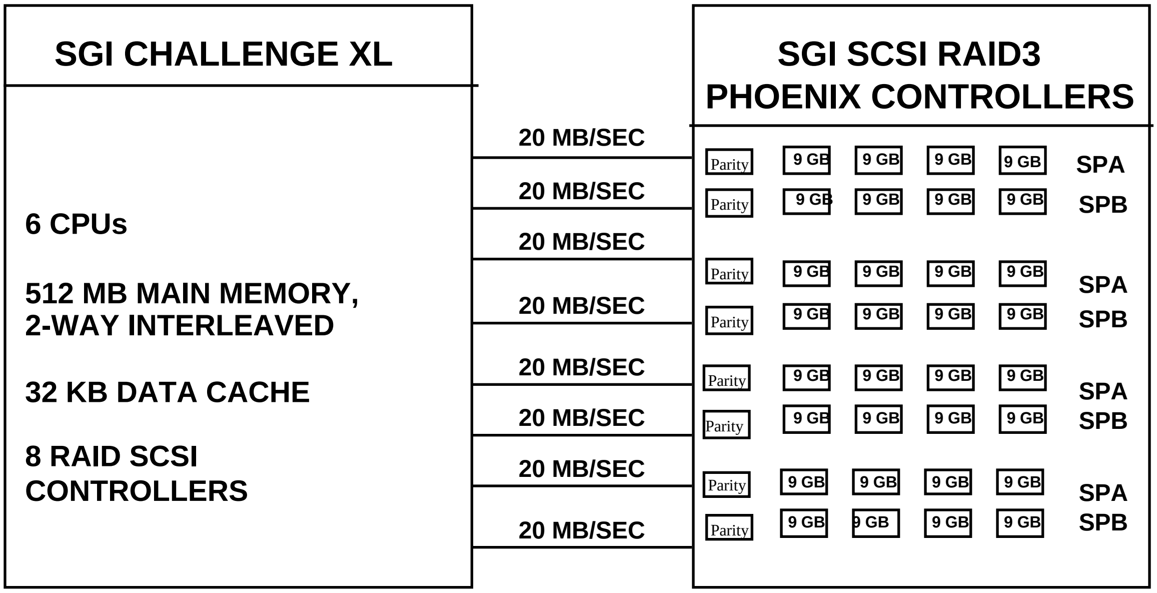

SGI CHALLENGE XL
6 CPUs
512 MB MAIN MEMORY, 2-WAY INTERLEAVED
32 KB DATA CACHE
8 RAID SCSI CONTROLLERS
20 MB/SEC
20 MB/SEC
20 MB/SEC
20 MB/SEC
20 MB/SEC
20 MB/SEC
20 MB/SEC
20 MB/SEC
SGI SCSI RAID3 PHOENIX CONTROLLERS
Parity 9 GB 9 GB 9 GB 9 GB SPA
Parity 9 GB 9 GB 9 GB 9 GB SPB
Parity 9 GB 9 GB 9 GB 9 GB SPA
Parity 9 GB 9 GB 9 GB 9 GB SPB
Parity 9 GB 9 GB 9 GB 9 GB SPA
Parity 9 GB 9 GB 9 GB 9 GB SPB
Parity 9 GB 9 GB 9 GB 9 GB SPA
Parity 9 GB 9 GB 9 GB 9 GB SPB

Figure A3.2-3 RAID Configuration Under Test as of 8/1997

Aside from physically enlarging the RAID configuration and supplementing it by four more SCSI RAID controllers, other RAID configuration parameters were adjusted as follows:

1. Command Tag Queuing (CTQ) was enabled on the RAID, with the CTQ depth set to 24;
2. the stripe element for both the raw and the file system portions of the disk was set at 1024 KB;
3. the raw and the file system portions were allocated each to a set of four separate RAID controllers;
4. in sequencing the SCSI address allocation to the A and B controllers or Storage Processors (SPs), during disk partitioning, the entire A side was listed first, then the entire B side, in order to minimize the detrimental effects of A/B interaction. Thus, when the A side is under load, the B side experiences the resonance effects but is not under load at that time and vice versa;
5. AMASS cache partitions were sequenced in a random order during AMASS software initial configuration, so that the load during data transfers is evenly distributed instead always falling on the same A/B SP partition pair first.

A.3.2.3.2 Software

AMASS use

1. Use of `dd` data copy. Very low initially throughput results can be partially attributed to the use of the UNIX copy, `cp`, command for data transfers to and from AMASS directories. Since the block size used by the cp command for data transfers is 4 KB, such transfers are slow. The `dd` command with a block size of 1024 KB is now used for all AMASS data transfers.

2. Appropriate tuning of AMASS cache. AMASS cache parameters were tuned to the prevalent file size in order to optimize recording of that file size to tape.

AMASS corrections

The following correction to AMASS were made by the vendor:

1. Adjustable size blocking factor to tape. Initially, the size of the data block written to tape was "hardwired" to 16 KB. The drive manufacturer, STK, recommends block size of 256 KB for maximum throughput. The bulk of performance improvement of a single tape drive throughput was realized after the introduction of configurable blocking factor to tape.

2. Asynchronous library operation. Initial design of AMASS did not allow for asynchronous library operation on mount and dismount. No library activity took place during the time when any one of the drives was performing a tape load, positioning, or rewind. Correction has been fully tested for the rewind and unload. The asynchronous tape mount correction was accompanied by time-out problems and is still under test.

3. Introduction of Asynchronous I/O. Asynchronous I/O contributes to raising the overall archive system performance.

A.3.3 HiPPI Performance

ECS tested the performance of HiPPI networks connecting SGI hosts during the period from December 1995 through May 1996.

Tests were performed between two SGI hosts connected via HiPPI through a HiPPI switch. The SGI hosts were Challenge L systems with R4400 processors; initial tests were performed with these machines running Irix 5.3, and later tests were performed with these machines running Irix 6.2.

Two test tools were used, ttcp and netperf. Both of these tools allow data to be sent from the memory of one host to the memory of the other via TCP/IP packet streams; hence the tools test the performance of the host processors, the underlying implementation of TCP/IP, and the networks.

Results (in megabytes transferred per second) were recorded as a function of the message size sent. Configuration parameters—principally the window size and the message transfer unit (MTU)—were also varied.

The initial results under Irix 5.3 showed transfers occurring at up to 55 MB/sec. These results were obtained for messages larger than about 16 KB; performance for smaller messages trails off to nil as the message size decreases, but performance for messages larger than 16 KB is essentially independent of message size. It was observed that performance improved as the MTU size was increased up to approximately 20 KB; beyond 20 KB (up to the operating system limit of 64 KB), performance was relatively constant with MTU size. Performance was observed to be a strong function of the window size, with performance increasing sharply as the window size was increased to 512 KB, the maximum supported under Irix 5.3.

The results above were for a single message transfer. When the message flow was made bidirectional (both hosts sending and receiving simultaneously), the aggregate throughput increased to about 65 MB/sec. When multiple bidirectional transfers were performed in parallel, a peak throughput of about 80 MB/sec was achieved.

In May 1996 the operating system on the hosts was upgraded from Irix 5.3 to Irix 6.2. The single message transfer test was repeated using an MTU of 64 KB and a window size of 1 MB. The observed transfer rate for message sizes larger than 16 KB was approximately 92 MB/sec, compared to 55 MB/sec under Irix 5.3. This was consistent with predictions made by SGI.

A.3.4 SGI RAID Performance

ECS tested the performance of SGI RAIDs during the period from April 1996 through July 1996.

Tests were performed using an SGI host configured with twelve 90 MHz R8000 CPUs, one gigabyte (GB) of RAM, and eight fast wide differential SCSI-2 channels. The SGI RAID array was configured with one RAID controller per SCSI-2 channel; each controller was configured with five 4.3 GB disks. Each group of five disks was bound as a RAID 3 group. Tests were run using two generations of SGI RAID controllers, the Sauna (old) and the Phoenix (new). An xfs filesystem was striped across the eight SCSI-2 channels.

An instrumented version of the UNIX `dd` command, called lmdd, was used as the test tool. lmdd allows the movement of data from memory to disk, from disk to memory, or from disk to disk. It captures performance statistics for these transfers. `lmdd` allows testing using both filesystem cached I/O and Direct I/O. UNIX filesystems normally cache disk I/O in memory buffers; this caching can improve performance if the data transfers are being requested in very small blocks, or if certain data are accessed repeatedly. This caching is typically done in block sizes of 4 KB, 16 KB, or 64 KB, depending on the vendor and the version of the operating system. The Direct I/O lmdd option allows these buffers to be bypassed. This reduces the overhead associated with each transaction, and makes it possible to access the disk devices using much larger block sizes. Because each transaction to a disk drive has a huge overhead—including moving the read/write heads—disk performance for small blocks (64 KB and less) is dominated by the overhead time, and performance is poor. Tests were run using Direct I/O block sizes from 64 KB to 4 MB.

Using the Sauna controllers, maximum read performance using eight controllers was 52 MB/sec, and maximum write performance was 11 MB/sec. Using the Phoenix controllers, maximum read performance using eight controllers was 81 MB/sec, and maximum write performance was 52 MB/sec. Performance in all cases improved as the block size was increased; as a rough rule of thumb, throughput with 4 MB block sizes was an order of magnitude better than throughput with 64 KB block sizes.

A.3.5 Ciprico Fibre Channel Performance

ECS tested the performance of Ciprico Fibre Channel arrays during a one week period in July 1996.

Tests were performed using an SGI host configured with four 150 MHz R4400 CPUs, two gigabytes (GB) of RAM, and one Prisa fibre channel interface. The Ciprico fibre channel array was configured with nine 9 GB disks. An xfs filesystem was configured on the array.

lmdd (see above in section A.3.4) was used to perform the tests.

The read performance of the array increased from 20 MB/sec with 64 KB blocks to 59 MB/sec with 4 MB blocks. Write performance increased from 11 MB/sec with 64 KB blocks to 61 MB/sec with 4 MB blocks.

A.3.6 BDS Performance

ECS tested the performance of Bulk Data Service (BDS) during July 1996.

BDS is a data transfer application from SGI specially tuned for large data transfers streamed over TCP/IP networks. It makes use of large packet sizes and efficient data transfer between the client and the application to significantly improve performance over standard protocols such as NFS and ftp. The BDS application essentially acts as a wrapper for standard nfs; when large data transfers are requested, it handles the transfers instead of the standard nfs daemon. BDS can be configured to be transparent to the client application—that is, to appear as NFS—or it can be explicitly invoked by the client.

Tests were performed between two SGI hosts, one equipped with 12 R8000 CPUs and one GB of RAM, and one equipped with 18 R8000 CPUs and one GB of RAM. One host was equipped with a RAID array having four Phoenix controllers; the other host had a RAID array with eight Phoenix controllers.

BDS was used to move files (via lmdd, see section A.3.4 above) from the filesystem having four controllers to the filesystem having eight controllers. The maximum read performance of the four controller filesystem was 48 MB/sec with one process and 57 MB/sec with eight concurrent read processes; the maximum write performance of the eight controller filesystem was 52 MB/sec with one process and 69 MB/sec with eight concurrent write processes. The file transfer using BDS was observed to occur at 27 MB/sec for a single transfer; when eight transfers were executed simultaneously, the aggregate transfer throughput was observed to increase to 48 MB/sec.

A.3.7 Compression Algorithm Performance

ECS tested the performance of two standard UNIX compression tools during April 1996.

At the user's request, ECS will compress ordered data before it is transmitted electronically to the user or before it is written to media. ECS will also use compression to reduce the network requirements to send data from one DAAC to another.

Tests were performed on an SGI host equipped with one R4400 processor; the processor clock speed was not recorded, but was most likely 150 MHz.

The standard UNIX utilities compress and gzip were tested. Each was tested with 1 MB, 10 MB, 50 MB, 75 MB, and 100 MB input files. gzip allows an optimization level to be set. At level 1, gzip produces a compressed file in the minimum processor time (for this algorithm); at level 10, it produces the smallest possible output file (for this algorithm), but uses significantly more processor time. Testing was done using levels 1, 5, and 10 of gzip optimization.

Input files of the appropriate size were generated by concatenating together library files (object code).

Using `compress`, the average compression was between 45% and 53%, and the resource utilization was between 1.4 and 1.7 seconds per megabyte (sec/MB) for compression and 0.6 and 0.8 sec/MB for decompression. Using `gzip -1`, the average compression was between 57% and 61%, and the resource utilization was between 1.5 s/MB and1.7 s/MB for compression and 0.4 and 0.5 sec/MB for decompression. Using `gzip -5`, the average compression was between 60% and 66%, and the resource utilization was between 2.9 sec/MB and 3.2 sec/MB for compression and 0.3 and 0.5 s/MB for decompression. Using `gzip -10`, the average compression was between 60% and 65%, and the resource utilization was between 19 sec/MB and 36 sec/MB for compression and 0.3 and 0.4 sec/MB for decompression.

The results indicate that `gzip -1` provides slightly better compression (smaller output files) and slightly less resource utilization (seconds per megabyte for compression and decompression) than `compress`. The higher levels of `gzip` compression are not cost effective, as they produce only minor improvements in the compression, and require major increases in resource utilization.

A.3.8 Autosys Performance

ECS tested the performance of Autosys during the fourth quarter of 1995.

Tests were performed on two Sun SPARC 20/50 machines equipped with 64 MB of RAM.

An Autosys plan was built to execute 1440 job boxes, each having four jobs. Each job caused a UNIX `sleep` to be executed, producing a delay between jobs. `sleep` was used to cause the minimal impact on the processing capacity of the test machines. The job boxes run by each client were made to be sequentially dependent on each other—the *N*th job box could not start until the (*N*-1)st job box was finished.

The total execution time (sum of the sleep commands) was two minutes for each box. With 720 job boxes per client, the expected duration of the test was 1440 minutes, or 24 hours. The test was observed to take 31 hours, with an average job box execution time of two minutes and 47 seconds.

The job box delays were analyzed. Some delay is due to the granularity of the UNIX `sleep` command. Other delays are caused by polling delays within the Autosys product. It was not possible to account for all of the delays; it was not clear that any of the delays were associated with a lack of system resources.

A.3.9 SQS And Illustra Performance

ECS tested the performance of Spatial Query Server (SQS) and Illustra during the period January 1996 through June 1997.

Tests were performed on an SGI Challenge L with eight 194 MHz R10000 processors and one gigabyte of RAM. The disk subsystem on the test machine consisted of three SGI RAID cabinets, each with two Sauna RAID controllers. Each RAID controller was configured on a separate SCSI-2 bus. Total storage in the array was 220 GB. The operating system on the server was Irix 6.2.

The database software tested was Sybase SQL Server 11.021, Sybase Open Client 10.0.3, SQS 2.2.2, and Informix Universal Data Server (UDS) 9.01.

A database was built for each product. Including data, indexes, and logs, the space required for each database was approximately 100 GB. Each granule record in the database was padded to require approximately 2 KB, as this is the expected size of a granule record in ECS.

Database performance was tested by running multiple client programs, each establishing a connection to the server and executing a sequence of queries. The queries were formulated to exercise the geospacial capabilities of the servers. A set of 24 basic query types was formulated. Each client cycled through the set of 24 query types; each client started at a different query in the cycle, and the data parameters used for each query type were varied by the clients. The result was a set of repeatable but quasi-random queries from the clients, having a predictable expected distribution.

Tests were run with three different numbers of simultaneous clients—20, 40, and 100. The clients were allowed to submit new queries for two hours, and the tests were allowed to run until all of the queries submitted in the two hour span had completed.

Many runs were executed, with performance tuning between runs performed by ECS personnel with assistance from the vendors.

UDS was observed to perform between 14 and 17 queries per minute. SQS was observed to perform between 4 and 5 queries per minute. UDS throughput was a factor of three to four times better than SQS throughput for each test.

A.3.10 PGE Performance

ECS collected performance data for many of the PGEs executed during Ir-1 SSI&T, using the rusage tool embedded in the ECS toolkit. This tool provides information about the clock time, processor time, memory, and I/O usage of the PGE.

Because the Ir-1 versions of the Science Software were considered to be immature, these data were not used for ECS design analysis.

ECS has also used the `pixie` code profiling tool available on the SGI to analyze the CERES SARB code integrated during Ir-1 SSI&T. This tool provides a count of the total machine instructions and total machine cycles used to execute a PGE, and also provides a break-out by code entry point. For each code entry point, it counts the number of times the entry point (subroutine) was called, the total instructions executed in that code segment, and the total number of machine cycles expended in that code segment. These statistics are an excellent starting point for a tuning effort because they identify where the system is using the resources required by the PGE. The ratio of machine instructions to machine cycles also indicates the degree of optimization already present in the instruction mix.

The profiling results for the SARB code were fed back to ESDIS and CERES. ECS will assist Instrument Teams in the use of `pixie` as a code profiling tool, but ECS does not currently have plans to profile the science code delivered by the Instrument Teams.

A.4 Performance Measurement and Tuning

Performance measurement activities through July, 1997 focused on testing the ingest-to-archive performance of ECS RC-2 software. This thread is documented in more detail in Appendix C (which discusses a later test of the same thread, performed in the mini-DAAC).

The tests were run at GSFC on July 30, 1997. Except for the AMASS software (which ran on an SGI machine, g0drg01), all ECS code ran on a single Sun machine (g0acs04). Three files, sized 100 MB, 500 MB, and 2 GB, were copied from g0acs04 to AMASS cache on g0drg01 and then to AMASS tape. In all cases, the copies concluded successfully, with transfer rates exceeding 1.5 x 10^6 bytes per second for each transfer.

An additional test was performed that day at GSFC to estimate the effects of sending multiple files simultaneously over the FDDI network. The client and server processes were written in perl. The server was multi-threaded. Network transfers used transmission control protocol (tcp) sockets. Perl sysread and syswrite functions were used, to ensure that no buffering would take place. The client generated the data directly (so no disk was involved on the client side); the data was sent across the network to the server; the server received the data and wrote it to local disk (AMASS cache). The client resided on a Sun machine (g0acs04) and the server reside on an SGI machine (g0drg01). The data rates observed were:

Number of simultaneous sends x size	Data rate per connection (MB/sec)	Total rate (MB/sec)
1 x 64 MB	2.4	2.4
2 x 64 MB	2.2	4.4
3 x 64 MB	2.0	6.0

This page intentionally left blank.

Appendix B. Performance Scenario Plan Template

Performance Scenario Name

Scenario Description

This paragraph provides a process-level description of the scenario in step-by-step fashion, movement of control between processes, and movement of data among computers and devices. Background or concurrent system workloads should be identified. If applicable, the initial system state should be characterized, e.g. to specify whether certain files have been pre-staged.

Performance Requirements/Goals

The applicable L3 performance requirements should be listed here as well as design goals. If requirements/goals differ from DAAC to DAAC, the entry for each DAAC should be listed. Expected outcomes of the test should be listed, also.

Process and Hardware Configurations

List of the ECS-developed processes to be run and the identifier of the processor on which each runs. Depending on where the scenario is to be run, processors from the mini-DAAC, the GSFC DAAC, or both may be listed here. Any relevant hardware configurations should be listed here, e.g. NFS version, RAID configurations, AMASS settings, etc.

Performance Data Collection

ECS Server Data

Describe what performance data is to be generated by the ECS server processes and how it is done (e.g. performance logging, screen capture). Identify where performance logging has been inserted in the code, what is logged, and the definition of the log formats.

System Data

Describe the system performance tools/scripts to be run, what data they are gathering, and at what time interval.

Scenario Data

Input Data

Description of the data listed by files required at the start of the scenario, including volume, content, format, and who is providing it.

Output Data

Description of data listed by files expected as output of the scenario and the method for determining its correctness.

Directories

Location for each of the scenario data files. If an AMASS tape archive is a source or destination of the data it should be noted including the processor on which the AMASS file system is running and the cache directory to be used.

Version of the Software

Identify the ClearCase® branch from which the tested software comes. Include internal version and patch identification for each module. Also identify the version of each piece of relevant commercial off-the-shelf (COTS) software.

Staff Resources

Test Personnel

List who will run the ECS servers

Performance Team

List who will run the performance data gathering scripts and who will do postprocessing of performance data

Support

List additional support staff required

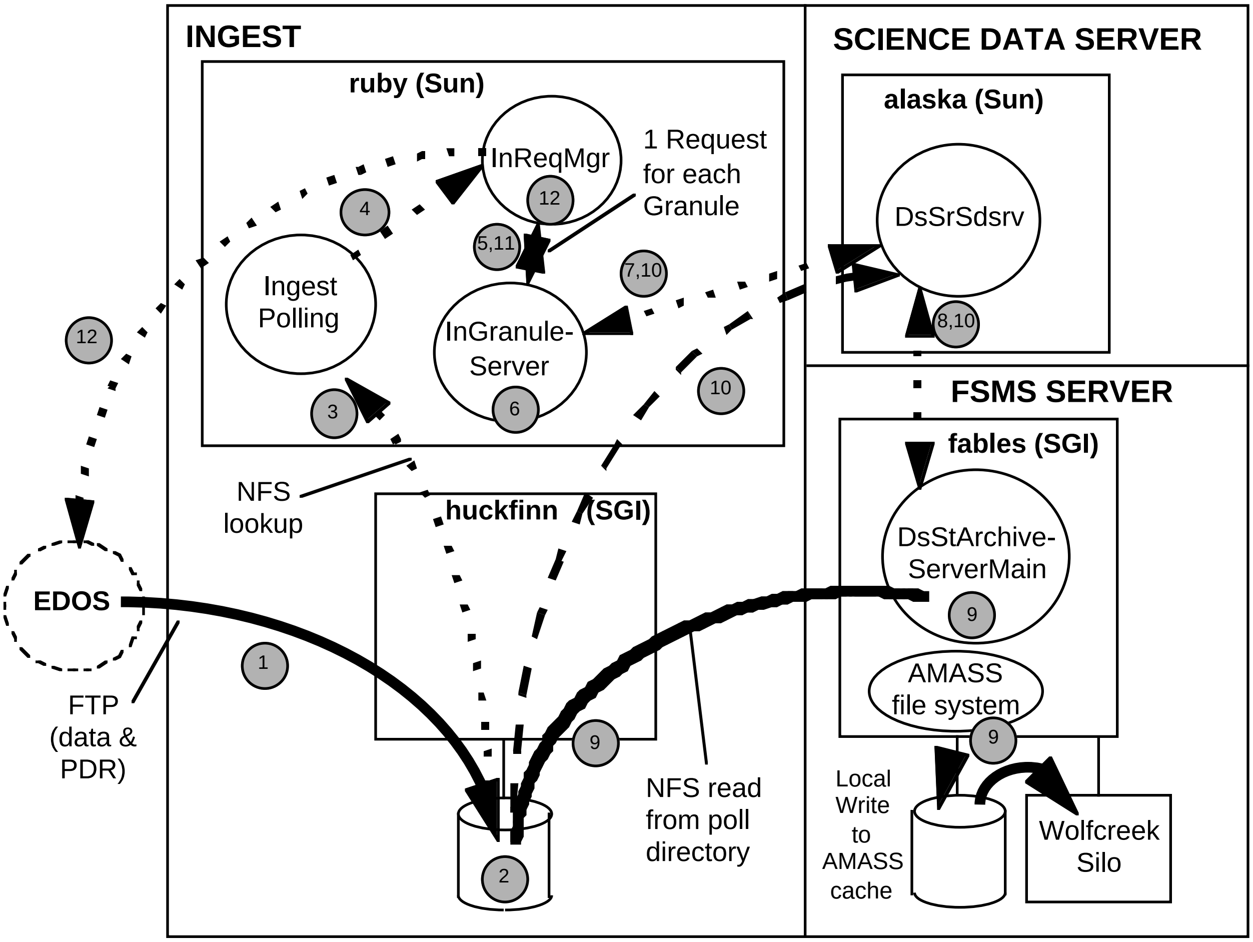
Schedule

List when the scenario is to be run (probably a reference to the master schedule in the plan). Estimate the amount of elapsed time expected for the test. Include a deadline for the post-processing work relative to the run date of the test.

Appendix C. Ingest-to-Archive Performance Scenario Plan

Scenario Description

The scenario is illustrated by the following diagram. The machines named (ruby, alaska, etc.) are in the ECS Landover Release B.0 Integration DAAC (the "mini-DAAC"). The numbered circles correspond to the sequence of events that follow the diagram.



1. Files are pre-staged to simulate the completion of an EOS Data and Operations System (EDOS) transfer of a granule set (data but no product description record (PDR)) to the ECS Internal Staging Disk via file transfer protocol (ftp).
2. The test begins when the PDR is copied into the polling directory of the Staging Disk.

3. The Ingest Polling Client, which is checking the contents of Internal Staging every 10 (start-up configuration defined) seconds, notices the PDR.
4. The Ingest Polling Client sends a PDR (in the form of an RPC) to the Ingest Request Manager.
5. The Ingest Request Manager figures out which Ingest Granule Server handles MODIS data and sends a request (RPC) to that instance. In a multi-granule ingest, a request for each granule is made simultaneously (granules will be ingested in parallel).
6. The Ingest Granule Server performs the metadata preprocessing.
7. The Ingest Granule Server then makes an Insert Request (Sync RPC) to Science Data Server to insert the granule.
8. Science Data Server makes a request (Sync RPC) to Storage Management to move the file from the input directory to the archive (AMASS cache directory).
9. Storage Management moves the granule file(s) by reading from the input directory and writing to the archive. Any files that are not local to the server on which Storage Management runs are accessed via NFS. Files within the same granule are moved sequentially.
10. When the transfer is done, Storage Management completes the RPC from Science Data Server, which in turn completes the RPC from Ingest and inserts the metadata into the Science database.
11. The Ingest Granule Server notifies the Ingest Request Manager after it completes processing each granule.
12. When all granules of this Ingest transaction are processed, the Ingest Request Manager marks the transaction complete and notifies EDOS.

Note: The test of this scenario will be conducted twice: once as described above and a second time with contention in the polling directory produced by an incoming FTP of 2 GB of data which begins at the same time as step 2.

Performance Requirements/Goals

The maximum at-launch requirement for ingest-to-archive throughput occurs at EDC; the needed rate is 3.0 megabytes per second. It is agreed between ESDIS and ECS that an appropriate goal for ECS in August, 1997 is to achieve a rate of 1.5 megabytes per second. The design goal for this scenario is 8 megabytes per second, which is the full achievable throughput of a FDDI network.

Process and Hardware Configurations

The following table provides the software to platform mapping:

Platform	Custom Executables, DBMS Servers, AMASS
ruby (SUN)	InGranuleServer InRequestManager InAutoIngest IngestGUI
huckfinn (SGI)	DsStStagingDiskServerMain DsStStagingMonitorServerMain DsStFtpIngestServerMain DsStArchiveServerMain Sybase Server
sorcess (SUN)	DsSt8mmStackerServer DsStD3TapeServerMain DsStFtpDisServerMain DsStPrinterServerMain DsDdRequestMgrMain
fables (SGI)	DsStStagingDiskServerMain DsStStagingMonitorServerMain DsStArchiveServerMain AMASS
journey (SGI)	Sybase Server
texas (SUN)	EcDssGUI DsStPullMonitorServerMain
alaska (SUN)	DsSrSdsrv

Performance Data Collection

ECS Server Data

Science data server logs via *EcUtPerfData* Object

Ingest Database tables `InRequestSummaryHeader.tbl` and `InRequestSummaryData.tbl`

Screen dumps with time tags

Periodic long list of the target directory (5 second interval)

System Data

Data will be collected via the following performance data processes on all processors listed above except as noted:

`sar` (60 second interval)

`sar -d` (60 second interval)

`nfsstat` (60 second interval)

`sysperf -k 15` (15 second interval, run on fables only)

Note: Care must be taken to ensure that only one instance of `sysperf` is running on fables. Multiple instances interfere with each other and produce bad data.

Scenario Data

Input Data

The data file and construction record making up the MODIS data set to be used as the input to the scenario are:

Size	Filename
1521993984	MOD_2GB.DATA
632	MOD.AM1.V1.hdr.L0.D1996216.000002to015459.D1997198

Output Data

Output data is copies of the input data sets in the Archive System.

Directories

Ingest Polling Directory: `/data4/OPS/icl/pollEDOS` on huckfinn

Storage Management Archive (AMASS cache): `/dss_stk1` on fables

Version of the Software

```
Branch = relb

Label = AUG_DEMO_V.1.0_BASE
```

Staff Resources

Development Personnel

Ernie Svehla, Jo Pulkkinen, Carey Gire, Ray Simanowith, Shankar Rachakonda, Dag Hestnes.

Performance Team

Larry Rapisarda—run the system data collection scripts

Chris Wilkinson—run the sniffer to monitor data traffic

Nick Singer and Larry Rapisarda—post processing and analysis

Schedule

Scenario was run on August 12, 1997. Analysis and documentation was completed August 15, 1997.

This page intentionally left blank.

Abbreviations and Acronyms

ACMHW	access control and management hardware
ACSLS	automated cartridge system library software
AHWGP	Ad Hoc Working Group on Production
AM-1	EOS AM Mission spacecraft 1, morning spacecraft series—ASTER, CERES, MISR, MODIS and MOPITT instruments
API	application programming interface
ARP	address resolution protocols
ASF	Alaska SAR Facility (DAAC)
ATM	asynchronous transfer mode
bds	bulk data service
CCR	configuration change request
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CDS	cell directory service
CI	configuration item
COLOR	Ocean Color - EOS Color Mission
COTS	commercial off-the-shelf (hardware or software)
CPS	contractor purchasing system
CPU	central processing unit
CSCI	computer software configuration item
CSMS	Communications and Systems Management Segment (ECS)
CSS	communication subsystem
CTQ	Command Tag Queuing
CUT	Code and Unit Test
DAAC	distributed active archive center
DBMS	database management system
DCCI	distributed computing configuration item

DCE	distributed computing environment (OSF)
DCHCI	distributed computing hardware configuration item
DFS	distributed file system
DID	Data Item Description
DIPHW	Distribution and Ingest peripherals hardware
DMS	Data Management Subsystem
DNS	domain name services
DPS	Data Processing Subsystem
DRPHW	Data Repository hardware
DSS	Data Server Subsystem
EBNet	EOS Backbone Network
Ecom	EOS Communications (replaced by EBNet)
ECS	EOSDIS Core System
EDC	EROS Data Center (DAAC)
EDF	ECS development facility
EDOS	EOS Data and Operations System
EGS	EOSDIS Ground System
EOC	Earth Observation Center; EOS Operations Center
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EP	evaluation package
ESDIS	Earth Science Data and Information System (NASA)
ESDT	Earth Science Data Type
ESN	EOSDIS Science Network (ECS) (replaced by EBNet)
F&PRS	Functional and Performance Requirements Specification
FDDI	fiber distributed data interface
FOS	Flight Operations Segment (ECS)
FSMS	File Storage Management System
ftp	file transfer protocol

GB	gigabyte (=10^9 bytes)
GDS	ground data system
GFE	Government furnished equipment
GSFC	Goddard Space Flight Center
GUI	graphical user interface
HiPPI	high performance parallel interface
HWCI	hardware configuration item
I&T	integration and test
ICLHW	Ingest client hardware
ICMP	Internet control message protocol
IDL	interface definition language
IDR	Internal Design Review
INCI	internetworking configuration item
INHCI	internetworking hardware configuration item
INS	Ingest Subsystem
I/O	input/output
IOS	Interoperability Subsystem
IP	Internet protocol
Ir1	interim release 1 (use Ir1 for EDHS search)
IRD	interface requirements document
ISS	internetworking subsystem
IST	Instrument Support Toolkit
IV&V	independent verification and validation
JPL	Jet Propulsion Laboratory (DAAC)
KB	kilobyte (=10^3 bytes)
LAN	local area network
Landsat	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LOC	lines of code

LPS	Landsat Processing System
M&O	maintenance and operations
MACI	management agent configuration item
MB	megabyte ($=10^6$ bytes)
MB/sec	megabytes per second
MCI	management software configuration item
MDT	mean downtime
MFLO	millions of floating-point instructions
MFLOPS	millions of floating-point instructions per second
mgmt	management
MHCI	management hardware configuration item
MIB	management information base
MLCI	management logistics configuration item
MSS	systems management subsystem
MTU	message transfer unit
MUI	management user interface
NFS	Network File System
NISN	NASA Integrated Services Network
NNTP	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration
NSI	NASA Science Internet
NSIDC	National Snow and Ice Data Center (DAAC)
ODC	other data center
OO	object oriented
OODCE	object oriented distributed computing environment
ORNL	Oak Ridge National Laboratory (DAAC)
OSF	Open Software Foundation
OSI	Open Systems Interconnection
PDL	program design language

PDR	Preliminary Design Review; product description record
PGE	product generation executive
PICS	procurement and inventory control system
PLS	Planning Subsystem
PMS	performance measurement system
POSIX	Portable Operating System Interface for Computer Environments
PPP	point-to-point protocol
PSCN	Program Support Communications Network
RAID	Redundant Array of Inexpensive Disks
RDBMS	Relational Database Management System
REL	release
RFC	request for comments
RFI	request for information
RFP	request for proposal
RIP	Routing Information Protocol
RMA	reliability, maintainability, availability
RPC	remote procedure call
SAGE III	Stratospheric Aerosols and Gas Experiment III
SCF	science computing facility
SDP	Science Data Processing
SDPS	Science Data Processing Segment (ECS)
SDR	System Design Review
SGI	Silicon Graphics, Incorporated
SLIP	serial line Internet protocol
SLOC	source lines of code
SMC	system monitoring and coordination center
SNMP	simple network management protocol
SPRHW	Science processing hardware
SQL	Structured Query Language

SQS	Spatial Query Server
SRS	software requirements specification
STL	Science and Technology Laboratory
TB	terabyte ($=10^{12}$ bytes)
TCP/IP	transmission control protocol/Internet protocol
TELNET	telecommunication network
TP	technical paper
TRMM	Tropical Rainfall Measuring Mission (joint US-Japan)
TSDIS	TRMM Science Data and Information System
UDP	user datagram protocol
UDP/IP	user datagram protocol/Internet protocol
WAN	Wide Area Network
WKSHW	Working storage hardware